CREATE A CHATPOT IN PYTHON

Phase 5 : Submission

Introduction

We will create an AI chatbot using Natural Language Processing (NLP) in Python. Our goal is to help you build a smart chatbot. First, we’ll explain NLP, which helps computers understand human language. Then, we’ll show you how to use AI to make a chatbot to have real conversations with people. Finally, we’ll talk about the tools you need to create a chatbot like ALEXA or Siri

# Introduction to AI Chatbot

As the topic suggests we are here to help you have a conversation with your AI today. To have a conversation with your AI, you need a few pre-trained tools which can help you build an AI chatbot system. In this article, we will guide you to combine speech recognition processes with an artificial intelligence algorithm.

**Speech to Text Conversation**

# Execute the AI

if \_\_name\_\_ == "\_\_main\_\_":

ai = ChatBot(name="Dev")

while True:

ai.speech\_to\_text()

# What is a Chatbot?

A chatbot is an AI-based software designed to interact with humans in their natural languages. These chatbots are usually converse via auditory or textual methods, and they can effortlessly mimic human languages to communicate with human beings in a human-like manner. A chatbot is arguably one of the best applications of natural language processing.

Chatbots can be categorized into two primary variants – Rule-Based and Self-learning.

**Installation**

Install chatterbot using Python Package Index(PyPi) with this command

pip install chatterbot

Below is the implementation.

# Import "chatbot" from

# chatterbot package.

from chatterbot import ChatBot

# Inorder to train our bot, we have

# to import a trainer package

# "ChatterBotCorpusTrainer"

from chatterbot.trainers import ChatterBotCorpusTrainer

# Give a name to the chatbot “corona bot”

# and assign a trainer component.

chatbot=ChatBot('corona bot')

# Create a new trainer for the chatbot

trainer = ChatterBotCorpusTrainer(chatbot)

# Now let us train our bot with multiple corpus

trainer.train("chatterbot.corpus.english.greetings",

"chatterbot.corpus.english.conversations" )

response = chatbot.get\_response('What is your Number')

print(response)

response = chatbot.get\_response('Who are you?')

print(response)

# Types of AI Chatbots

Chatbots are a relatively recent concept and despite having a huge number of programs and NLP tools, we basically have just two different categories of chatbots based on the NLP technology that they utilize. These two types of chatbots are as follows:

Scripted Chatbots

Scripted chatbots are chatbots that operate based on pre-determined scripts stored in their library.

# cleaner.py

import re

def clean\_corpus(chat\_export\_file):

message\_corpus = remove\_chat\_metadata(chat\_export\_file)

cleaned\_corpus = remove\_non\_message\_text(message\_corpus)

return cleaned\_corpus

# ...

# Deleted: if \_\_name\_\_ == "\_\_main\_\_":Chatbots

#### Artificial intelligence

Artificially intelligent chatbots, as the name suggests, are designed to mimic human-like traits and responses.

### Installing Packages required to Build AI Chatbot

We will begin by installing a few libraries which are as follows :

# Code:

# To be able to convert text to Speech

! pip install SpeechRecognition #(3.8.1)

#To convey the Speech to text and also speak it out

!pip install gTTS #(2.2.3)

# To install our language model

!pip install transformers #(4.11.3)

!pip install tensorflow #(2.6.0, or pytorch)

We will start by importing some basic functions:

import numpy as np

We will begin by creating an empty class which we will build step by step. To build the chatbot, we would need to execute the full script. The name of the bot will be “ Dev”

# Beginning of the AI

class ChatBot():

def \_\_init\_\_(self, name):

print("----- starting up", name, "-----")

self.name = name

# Execute the AI

if \_\_name\_\_ == "\_\_main\_\_":

ai = ChatBot(name="Dev")

Output :

building the AI for Chatbot

**CONCLUSION**

In this project, we have introduced a chatbot that is able to interact with users. This chatbot can answer queries in the textual user input. For this purpose, AIML with program-o has been used. The chatbot can answer only those questions which he has the answer in its AIML dataset.